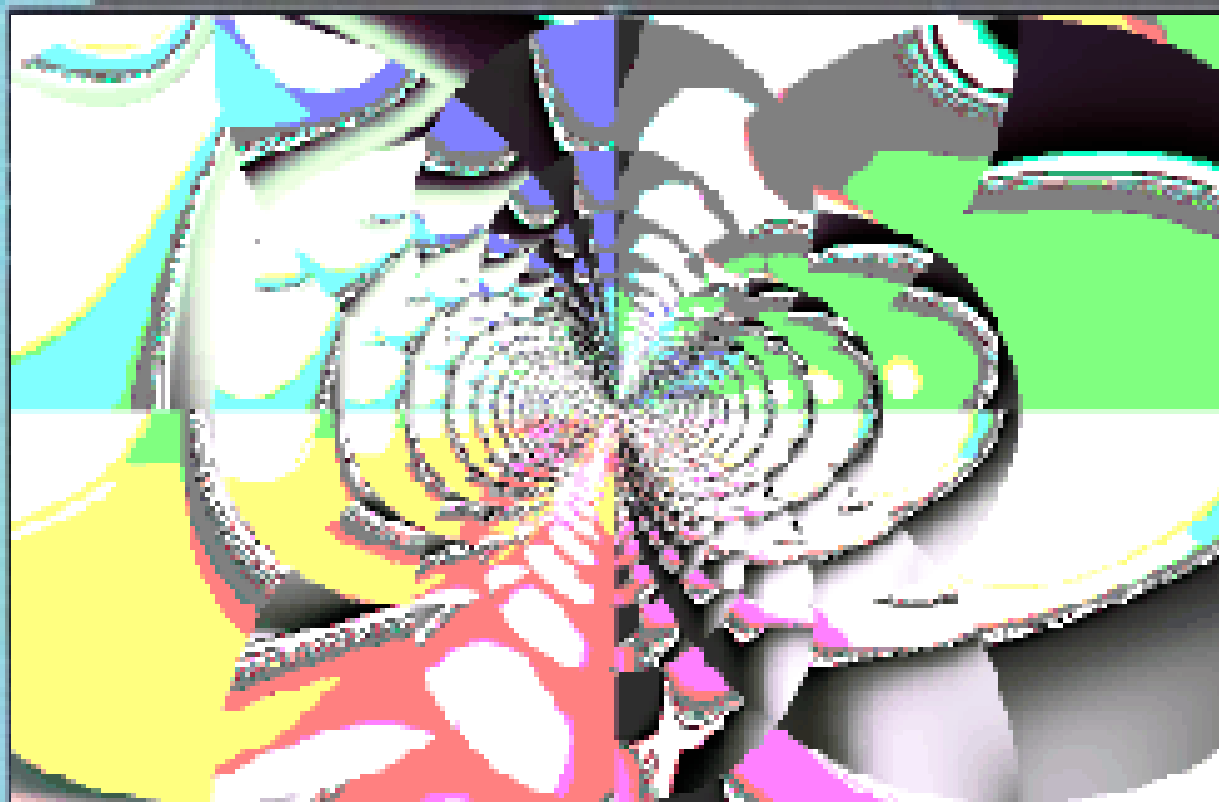


Boris
Petkoff



Business & Computing

MetaDesign (R)evolution

DomainOntology
ComponentLibrary
ApplicationFactory



MetaDesign (R)evolution

VORWORT	10
1 PROLOG - METADESIGN	17

OntoLogik

2 DIE ONTOLOGISCHE PERSPEKTIVE	32
2.1 Metaphysik	33
2.2 Ontologie	34
2.3 Epistemologie	35
2.4 Wissenschaftstheorie	36
3 DIE TRADITIONELLE UND FORMALE LOGIK	42
3.1 Die Lehre vom Begriff	43
3.2 Die Lehre vom Urteil	49
3.3 Die Lehre von den Schlüssen	52
3.4 Formale Logik	55
3.5 Abbildungen, Transformationen und Modelle	59
4 WAHRHEITSTHEORIEN	68
4.1 Kohärenztheorie vs. Korrespondenztheorie	70
4.2 Semantische Definition der Wahrheit	72
4.3 Sprache und Wirklichkeit	75
4.4 Konsenstheorie	76
5 ALLGEMEINE SYSTEMTHEORIE UND KYBERNETIK	79
5.1 Das Bertalanffy-Programm	81
5.2 Grundbegriffe und Methoden	85

5.3	Differenz	86
5.4	System	87
5.5	System und Umwelt	87
5.6	Reduktion von Komplexität (Negentropie)	88
5.7	Funktion und funktionale Analyse	91
5.8	Ordnung, Organisation und Struktur	92
5.9	Operation und Prozeß	93
5.10	Code und Information	94
5.11	Kybernetik und Rückkoppelung	96
5.12	Zweck, Teleonomie und Homöostasis	96
5.13	Funktionale Differenzierung, Subsysteme und Komplexität	97
5.14	Evolution, Transformation, Selbstorganisation und Emergenz	98
5.15	Kybernetik und Management	100
5.16	Viable System Model	102
5.17	Architektur und Regelwerk	103
5.18	Diagnose, Regulierung und Design komplexer Systeme	106

OntoGenetik

6	ONTOGENETISCHE ASPEKTE	109
6.1	Die soziale Analyseebene	127
6.2	Experiential Learning Model	128
6.3	Die strukturelle Analyseebene	131
7	EMPIRISCHE THEORIEMODELLE UND SYSTEME	137
7.1	Erfahrungswissenschaftliche Theorien	140
7.2	Theorieelemente	140
7.3	Intendierte Systeme	143

7.4	Strukturalismus	143
7.5	Approximation	144
7.6	Computerbasierte Metamodellierung	145
7.7	Basic ontological assumptions	147
8	SEMANTISCHE KONZEPTION VON THEORIEMODELLEN	148
8.1	Strukturen	149
8.2	Modelle	152
8.3	Potentielle Modelle	152
8.4	Empirische Systematisierung	153
8.5	Theoretizität	153
8.6	Partielle potentielle Modelle	155
8.7	Intendierte Systeme	156
8.8	Messung	157
8.9	Empirische Behauptung	158
8.10	Strukturalistische Rekonstruktion	159
8.11	Theorienetze	161
8.12	Spezialisierung	162
8.13	Theoretisierung	163
8.14	Reduktion	164
8.15	Querverbindungen	165
8.16	Theorie-Holone	167
9	REPRÄSENTATION UND HANDLUNGSTHEORIE	169
9.1	Reflexion und Intentionalität	169
9.2	Handeln und Kommunikation	170
9.3	Begriffsnetze und Propositionen	171
10	PROBLEMLÖSEN MIT THEORIENETZEN	176

10.1	Planen in Theorienetzen	178
10.2	Konflikte	181
10.3	Zusammengesetzte Konflikte	185
10.4	Formale Theorien	186
11	ONTOGENETISCHE ANSÄTZE	188
11.1	Methodologische Systematisierung	189
11.2	Analogie-Modelle	192
11.3	Laws of Form	196
11.4	Konzeptuelle Modelle, und Modelltransformationen	207
11.5	Ontologien vs. Metamodelle und Referenzmodelle	209
11.6	Persönliche Konstrukte	215
11.7	Formale Begriffsanalyse	218
11.8	Begriffliche Graphen	222
12	ACCORD METHODOLOGIE	226
12.2	ACCORD als Selbstorganisationsstruktur	237
12.3	ACCORD als generischer Rahmen	244

OntoEngineering

13	ONTOENGINEERING ANSÄTZE	251
13.1	Sprachen und Generationen.	254
13.2	Niedere Programmiersprachen	254
13.3	Höhere Programmiersprachen	255
13.4	Datenbank- und Anwendungssprachen	255
13.5	Objektorientierte Programmiersprachen	256
14	OO-PARADIGMA MERKMALE	259

14.1	Abstraktion	259
14.2	Klassen und Objekte	260
14.3	Datenkapsellung	260
14.4	Vererbung	261
14.5	Wiederverwendbarkeit	261
14.6	Polymorphie	262
14.7	Vorgehensmodelle bei der OO - Softwareentwicklung	262
14.8	Anwendungsfälle und BASE	266
15	VON MODELLEN ZUR ANWENDUNG	278
15.1	Strukturalistische Konzeptualisierung	279
15.2	Problemlösen mit AgentenNetzen	283
15.3	Verteiltes Problemlösen	285
15.4	Die Multi-Agenten Perspektive	292
15.5	Planen in Multiagenten-Systemen	296
16	INTERNET UND CLIENT/SERVER	300
16.1	Plattformen für verteiltes Problemlösen	301
16.2	Transaktionen	303
16.3	Server-Aspekte	304
16.4	Middleware	307
17	COMPONENTEN PARADIGMA	309
17.1	Begriffsdefinition Komponentenbasierte Anwendungssysteme	312
17.2	Komponentenbasierte Softwareentwicklung	313
17.3	Die Legosteine-Metapher	316
17.4	Flexible Verteilung und Anwendungsnahe	317
17.5	Schnittstellenspezifikation und Implementierung	322
17.6	Programmierung von Applikationen mit BC4J	324

17.7	Migration zu komponentenbasierten Softwaresystemen	327
18	OBJEKTRELATIONALER DATENBANKDESIGN	332
18.1	Die Bedeutung von Datenbankmodell und -design	333
18.2	Das Objektmodell und das Datenbank-Design	337
18.3	Regeln für die Abbildung Objektmodells auf relationale Implementierung	340
18.4	Die Architektur von Java Database Connectivity (JDBC)	349
18.5	Typen von JDBC-Treibern	351
18.6	Die JDBC-Treiber von Oracle	351
19	DIE BC4J-FUNKTIONALITÄT	354
19.1	BC4J - Konzeptuelle Modelle und Modelltransformationen	356
19.2	BC4J - alternative Einsatzbereiche	363
20	VERTEILTEN ANWENDUNGEN: J2EE ODER .NET	375
20.1	Unterschiede zwischen J2EE und .NET	378
20.2	Visual Studio .NET	379
20.3	Web Services	385
20.4	Web Portale	390
20.5	Das ACCORD-Portal	402
20.6	Enterprise Application Integration (EAI)	409
20.7	House of Business Engineering	415
21	BUSINESS CAPABILITY MATURITY MODEL	423
21.1	Level 0 – zufällig	425
21.2	Level 1 – definiert	425
21.3	Level 2 – optimiert	426
21.4	Level 3 – gesteuert	426
21.5	Level 4 – gemeistert	427

OntoManagement

22	ACCORD UND GENERALPLAN VON UNTERNEHMEN	429
22.1	Modellierung von Organisationen und Softwaresystemen	432
22.2	Orientierung im Generalplan-Industrieunternehmen	434
22.3	Orientierung und Identifizierung der Funktionen	438
22.4	ACCORD und GPS Software Atlas	444
23	ORGANISATORISCHER WANDEL	449
23.1	Technologie Wandel	451
23.2	Garantiefall im Internet Shop	455
23.3	Informationstechnologie als Garant für Unternehmenserfolg?	460
23.4	Produktivitätsparadoxons der Informationstechnik	461
23.5	Wirtschaftlichkeitspotentiale	463
24	ARCHITEKTURRAHMEN FÜR GESCHÄFTSPROZEOPTIMIERUNG	469
24.1	Der Zyklus zum organisatorischen Wandel	469
24.2	Problemorientierte Differenzierung des abduktiv-deduktiven Zyklus	471
24.3	Anwendung des ACCORD-Modells im Handel	476
24.4	Methodische Anwendung des ACCORD-Vorgehensmodells	483
25	DIGITALE FABRIK	487
25.1	IT im Unternehmen	489
25.2	Entwicklungsarten	492
25.3	Komplexität = ProduktOntologie x FabrikOntologie	493
25.4	Kriteriensystem und Technologiebewertung	497
25.5	Ontologiebasiertes Konzeptdesign	502
25.6	Quality Function Deployment (QFD)	505
25.7	FehlerMöglichkeiten und EinflußAnalyse (FMEA)	513

25.8 Vorgehensmodell	513
26 EPILOG: SEMANTIC WEB	520
LITERATUR	526

Vorwort

Was wir heute World Wide Web nennen, ist manchmal eine - π x Daumen Kultur - oft nur ein primitives Flickwerk alter Technologien das tonnenweise bedeutungslosen Inhalten transportiert. Der Besucher von einer Firmenwebsite könnte zwar den Eindruck gewinnen, dahinter stehe ein einheitliches high-tech System. Doch die Mehrheit der Unternehmen, die Internetgeschäfte betreiben, haben ihre vorhandenen Informationssysteme nur mit neuen zusammengelegt - die Integration ist vorgetäuscht. Im Hintergrund arbeiten meist Mitarbeiter, die Informationen manuell von einer Anwendung auf eine andere übertragen. Dieses Vorgehen ist ineffizient, langsam und fehleranfällig. John McKinley, Chief Technology Officer von Merrill Lynch, bezeichnet die heutigen Rechnerwelten als Potemkinsche Dörfer, in denen bunt gestrichene Fassaden die heruntergekommenen Häuser verbergen.

Man hat die New Economy sorgfältig untersucht, seit 10 Jahren ihr die größte Aufmerksamkeit entgegengebracht, mit allen Mitteln ihr Wachstum gefördert und deren Keime „gehegt und gepflegt“. Es war unvermeidlich, dass einige von den DotCom's zugrunde gehen. Darauf kommt es nicht an. Worauf es aber ankommt nach dem Börsen Crash der New Economy ist eine Analyse der Aspekte, die das Überleben der lebensfähigsten sichern können. Diese Analyse muß sowohl den Inhalt, als auch Hilfsmittel und Strategie der New Economy betreffen. Das Wirtschaftssystem bildet in seinem Ablauf einen Prozeß, der sich auf mindestens zwei verschiedenen Ebenen analysieren lässt:

- auf der *strukturellen* Ebene des Wissens, wie es in den neuen Medien dargestellt und sich im Laufe der Zeit ändert und entwickelt. Solches Wissen kann als Produkt, als „Output“ angesehen werden. Es dient als „Rohstoff“ oder „Input“ für andere gesellschaftliche Teilsysteme: Technik, Wirtschaft, Politik.
- auf der Ebene der Menschen, ihrer Handlungen, Ziele, Werte, der Ebene der sozialen Gruppen und Institutionen, kurz: der *sozialen* Ebene.

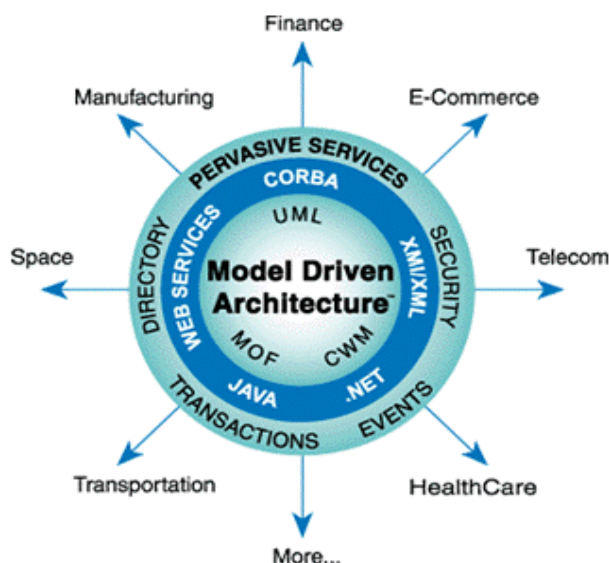
Eine Theorie, die all diese Aspekte berücksichtigt, ist zur Zeit nicht mal in Ansätzen absehbar, obwohl die Wissensbasis der Informatik, Psychologie, Soziologie, Philosophie, Linguistik u.a. schon reichhaltig und komplex genug ist, um eine theoretische Systematisierung zu rechtfertigen. Aus diesen Gründen werden wir versuchen ein MetaDesign-Ansatz¹ für Collaborative MultiAgent Systeme (cMAS) theoretisch zu begründen. Aus diesem Grund werden wir in Kenntnis der Chronologie sowohl der Epistemologie der Erfahrung, der Führungstheorien, als auch der Computer Science einen paradigmatischen Weg beschreiten, der uns neue Dimensionen für den Begründung und Entfaltung einer holistischen Theorie des MetaDesigns von Collaborative MultiAgent Systeme (cMAS) eröffnet.

Anhand des MetaDesign-Ansatzes werden wir versuchen, - neben der Abgrenzung von einer Cyber-Space Euphorie, geprägt durch Marketing und die Betonung von Optik gegenüber Substanz, - das notwendige und hinreichende Know How unter den Aspekten Acquisition, Design, Engineering und Management im Rahmen einer New Economy festzuhalten. Das vorgefundene ideologische und technologische Chaos zwingt uns die zu behandelnden Fragen ähnlich wie bei der Object Management Group - Model Driven Architecture (MDA), unabhängig von existierenden Technologien zu stellen, aber gleichzeitig umfassender. Weil mit der erkenntnistheoretischen Beschreibung des MetaDesign-Ansatzes gleichzeitig die Voraussetzungen geschaffen werden sollen, in diesem Zusammenhang grundsätzliche Fragen wie das Verhältnis von Schein und Sein zu klären, z.B.: ist New Economy elektro-

¹ Wir versuchen ein MetaDesign-Ansatz, im Einklang mit den Ideen von Youngblood und Maturana zu entwickeln, das nichts mit den gleichnamigen, - zwischen Berlin und Bangkok angesiedelten – WebWerbeAgenturen, zu tun hat.

nisch Plakate kleben, Litfaßseulen drehen, Kioske beliefern, Basare eröffnen oder Umschlagplatz für Wissen und Beratung, high-tech Produkte, intelligente Dienstleistungen und konstruktivistische InternetDidaktik. Wir brauchen Langlebigkeit, Wiederverwendung d.h. Transparenz.

Deswegen ist auch die OMG-Strategie MDA (Model Driven Architecture) zur modellgetriebenen Soft- und Hardwareentwicklung, eine Strategie die klare Trennung von Abstraktionsschichten bei der Modellierung von Systemen die **Wiederverwendbarkeit /Langlebigkeit der Modelle** sichert. Das **Ziel** der MDA ist es die momentan oft noch herrschende Lücke zwischen Modellen (bzw. Modellierern) und Quelltext (bzw. Programmierern) zu schließen - wobei die OMG durch ihre frei erhältlichen Standards die dazu notwendige Integration von verschiedensten Werkzeugen unterschiedlicher Hersteller ermöglicht, was für eine breite Akzeptanz sorgen dürfte. Um die Langlebigkeit der Modelle zu sichern ist es wichtig, dass sie unabhängig von einer bestimmten Plattform modelliert werden. Daher ist eine allgemeingültige Plattformdefinition das zentrale Kernelement der Spezifikationsbemühungen und derzeit Inhalt intensiver Diskussionen innerhalb der OMG.



OMG Model Driven Architecture™

How Systems Will Be Built

MDA™ provides an open, vendor-neutral approach to the challenge of interoperability, building upon and leveraging the value of OMG's established modeling standards: Unified Modeling Language™ (UML™); Meta-Object Facility™ (MOF™); and Common Warehouse Meta-model™ (CWM™). Platform-independent Application descriptions built using these modeling standards can be realized using any major open or proprietary platform, including CORBA®, Java, .NET, XMI™/XML, and Web Base platforms.

Vergegenwärtigen wir uns noch einmal die Entwicklung: Anfang der sechziger Jahre kamen die ersten Programme in Binärcode auf Lochstreifen auf; heute Ende der neunziger gehören Java Applets, Servlets, Portlets und Internet zu den Selbstverständlichkeiten der Computerwelt. Analysiert man gleichzeitig, wie Menschen innerhalb dieses Zeitraums über Computer dachten und mit Computern umgingen, kommt man nicht umhin, sich des durchgehenden Einflusses eines mechanistischen Weltbildes bewußt zu werden. Ein Weltbild das besonderen Wert auf *Daten, Information, Wissen* und *Intelligenz* legt, aber unfähig ist, selbst eine Definition, geschweige denn eine Erklärung dieser Denkkonstruktionen zu geben. In dieser naiv-rationalistischen Tradition ist ein Großteil des heutigen technologischen Fortschritts begründet, aber auch die unreflektierten Mythen von denkenden Computern, die ihrerseits die Grundlage für viele gegenwärtige Probleme und Irritationen gelegt haben. Selbst die Metaphern, die in die Welt gesetzt werden, was Computer leisten oder nicht leisten können, zeugen von einer eigentümlichen Blindheit gegenüber dem Wesen menschlichen Denkens, Entdeckens und Erfindens, von einer Blindheit, die zu einem weitverbreiteten Mißverständnis über die

von einer Blindheit, die zu einem weitverbreiteten Mißverständnis über die zukünftige Rolle von Computern führen kann.²

Bezüglich populär gewordener Spekulationen muß festgestellt werden:

- Hoffnungen auf eine Vollautomatisierung der menschlichen Tätigkeit, die mit der sogenannten "Künstlichen Intelligenz" verbunden werden, sind utopisch, da die Möglichkeiten der heutigen Automaten eng mit der Formalisierung der modellierten bzw. simulierten Informationsprozesse verbunden sind.
- Dennoch sind die Potenzen der modernen Computer nicht auf schematische Routineaufgaben begrenzt. Das wird bei der Analyse der schon existierenden und sich in der Entwicklungsphase befindenden Konzeptionen für wissensbasierte Systeme deutlich, die zukünftig eine organische Verbindung der maschinell realisierten Operationen mit den nicht-formalisierbaren schöpferischen Prozessen erlauben könnten.

Um die Erscheinungsformen einer neuen Technologie begreifen zu können, muß man die Frage nach dem Design-Ansatzes, d.h. nach dem Zusammenspiel von Verstehen und Herstellen stellen. Wenn in diesem Zusammenhang von Design die Rede ist, kann man sich nicht auf die Methodik planmäßigen Entwerfens beschränken. Man muß die umfassendere Frage stellen, wie eine Gesellschaft Erfindungen hervorbringt, deren Umsetzung wiederum die Gesellschaft insgesamt verändern. Man muß eine theoretische Grundlage entwickeln, um zu untersuchen, welche globale Wirkung die weltweit- vernetzten computerbasierte Systeme haben, nicht nur, wie sie funktionieren.

Zur Ausarbeitung einer solchen theoretischen Grundlage muß man einen Schritt zurückgehen und das implizite Verständnis von Gestaltung untersuchen, das als Teil der bestehenden Denktradition technologische Entwicklungen steuert. Nur durch Aufdecken dieser Tradition und durch explizites Bewußtmachen ihrer Hintergrundannahmen kann man sich für Alternativen und für sich daraus ergebende neue Gestaltungsmöglichkeiten öffnen.

Diese Art zu hinterfragen, kann man mit der Frage *Was ist ein collaboratives Multiagenten System?* deutlich machen. Als erstes ist zu beachten, daß aus unterschiedlichen Interessen verschiedener Individuen jeweils unterschiedliche Antworten erwachsen. Für den Geschäftsführer eines Softwarehauses, das solche Systeme entwickelt, ist ein collaboratives MultiAgenten-System eine problemorientierte Konfiguration von SoftwareComponenten zur Eingabe, Speicherung und Ausgabe von Information, die programmiert, getestet und ausgeliefert werden müssen. Für die Person, die das collaborative MultiAgenten-System designt, ist es eine Herausforderung, innovative Architekturen für Software-Agenten zu entwerfen und Wissensakquisitions-, Wissensrepräsentations- und Inferenzkomponenten bei der Implementierung einzusetzen.

Obwohl von einer Theorie der collaborativen MultiAgenten-Systeme nicht die Rede sein kann, ist es seit Jahren schick über innovative **Belief-Desire-Intention**-Architekturen zu reflektieren. Die erste Frage, wenn man ein neues Modell einführt, wäre: „Ist es zu viel des Gutem oder zu wenig?“ bzw. „Warum **Belief-Desire-Intention** und nicht **Liebe-Glaube-Hoffnung**?“ Diese Frage würde jeder Informatiker als eine Provokation empfinden und bestenfalls auf die so genannten Theorie des **”praktischen Denkens**“, verweisen, die es leider gar nicht gibt. Zumindest im wissenschaftstheoretische Sinn des Wortes. Wenn man aber weiter einfache Fragen stellt, wie z.B. „Was versteht man unter Theorie

² Petkoff, Boris: Wissensmanagement, Addison Wesley, 1998

?“ hat man den Beweis erbracht, keine Ahnung zu haben, welche Fragen in Informatik-Kreisen zulässig sind und welche nicht. Die Antwort wäre:

- in der Java-Welt: „null pointer exception“;
- in der KI-Welt dank “closed world assumption”: „not true“;
- auf Deutsch: „Suche nicht nach Sinn, dort wo keiner ist!“

Wir werden in Zusammenhang mit dem MetaDesign-Ansatz solche Fragen beantworten müssen. Das schalten von Bits und Bytes und das ermitteln des Wahrheitswertes einer logischen Formel, bringen uns nicht weiter. In der „open WorldWideWeb“ hat die “closed world assumption” nichts zu suchen. Mit der globalen digitalen Vernetzung sind neue Formen der Kommunikation, der Information und des Handelns entstanden, die auf der Nutzung der Computernetze als digitalen Medien aufbauen. Die Gutenberg’sche Ordnung des Wissens, die vor allem entlang des Textes und des Drucks entfaltet wurde, gerät nun in den Sog der offenen, globalen Rechnernetze. Die national geprägten Wissensordnungen werden von einer neuen globalen Wissensordnung abgelöst. Die Zukunft der Wirtschaft wird entscheidend durch seine beschleunigte digitale Technisierung geprägt.

Wir müssen den Versuch unternehmen, die Kluft zwischen den strategischen Reflexionen der Old Economy und der operativen Verdrahtung der Geschäftsprozesse der New Economy zu überbrücken, d.h. eine systemische MetaDesign-Plattform für *continuous engineering* mit rückgekoppelten Übergängen zwischen strategischen, taktischen und operativen Entscheidungsvorgängen zu definieren, die uns die transparente computerbasierte Entscheidungsunterstützung erlaubt. Im Kontext dieser Abhandlung gilt es, die Aspekte dieser neuen Wissensordnung herauszuarbeiten und diese zu einem MetaDesign-Ansatz zu verdichten, der Rechnernetze als Wissensmedien angemessen beschreibt. Da eine Theorie des Wissens als Referenzmodell trotz 50 Jahren KI-Forschung nicht mal in Ansätzen existiert, verstehen wir unter angemessen holistisch, d.h. ontologisch, erkenntnistheoretisch, wissenschaftstheoretisch, kybernetisch, betriebswirtschaftlich, informationstechnisch, etc.

Das Hauptproblem ist, eine epistemologische Brücke zwischen scheinbar inkompatiblen betriebswirtschaftlich-strategischen und informationstechnisch-operativen Konzepten zu schlagen, um das Operationalisierbare beider Aspekte auf einen gemeinsamen Nenner zu bringen. Die Aussichten auf Erfolg bei der Suche nach den Gemeinsamkeiten sind nicht besonders gut, da sie in den gemeinsamen epistemologischen Wurzeln dieser Fachrichtungen liegen, und die sind philosophischer Natur. Zudem muß man computerbasierten Fachrichtungen nicht nur ein gestörtes Verhältnis zur Philosophie, sondern sogar zu Theorie nachsagen.

Denn seit Ende der 80er Jahren wird Software nicht nach den theoretisch-begründeten Ideen von Hochschullehrer wie z.B. der ETH Prof. N.Wirth hergestellt, sondern die ad-hoc Implementationen von Softwarehäuser wie SAP&Co werden von Hochschullehrer manchmal als „state of the art“ der Lehre vermittelt, frei nach dem Motto: die Erfolgreichen sind die Besten, denn nur die Besten haben Erfolg. Hier wird im Endeffekt das Normale zur Norm erhoben, das Machbare zum Maßstab rationalen Vorgehens gemacht. Eine solche Erhebung des Normalen zur akzeptierten Norm ist in methodologischen Fragen immer verhängnisvoll: Sie beraubt uns des kritischen Maßstabs für die Reflektion auf die Täuschungsquellen im Normalen, auf dessen normativen Gehalt.

Da die Kenntnissen der Informatiker im Normalfall als einseitig und fragmentarisch zu bewerten sind, erschien eine sequentielle Struktur ratsam, die das „notwendige und hinreichende“ Wissen, um ein reales Problem mit einer n-Tier Applications zu lösen, von A bis Z thematisiert, d.h. DomainOntology,

UML-KlassenModelle, UML-ProzessModelle, DomainBusinessRules-Tier, GUI Client-Server, Web Client-Server, Web Services, Web Portale, Enterprise Application Integration etc.:

1. Kapitel – METADESIGN: thematisiert die für das Thema MetsDesign bedeutenden Traditionen bzw. Weltbilder, da jede Modellierung auf der Basis tiefgreifender Annahmen über die Realität betrieben wird, und damit die problemrelevanten Dimensionen des Vorgehens, d.h. der Rahmen des Ansatzes vorgibt.
2. Kapitel – DIE ONTOLOGISCHE PERSPEKTIVE: thematisiert die Metaphysik, Ontologie, Epistemologie, Wissenschaftstheorie als philosophische Aspekte in der Tradition der problemrelevanten Reflexion über Realitätsausschnitten.
3. Kapitel – DIE TRADITIONELLE UND FORMALE LOGIK: thematisiert die heute noch allgemeingültigen Aspekte der traditionellen Lehre vom Begriff, vom Urteil, von logischen Schlüssen und ihre zwecks Automatisierung verkürzten formallogischen Ausprägungen wie Abbildungen, Transformationen und Modelle.
4. Kapitel – WAHRHEITSTHEORIEN: thematisiert die existierenden Wahrheitstheorien – Kohärenz, Korrespondenz-, Konsenstheorie und Semantische Definition der Wahrheit, wie Sprache und Wirklichkeit, - die implizit das wirksame Weltbild bestimmen, unabhängig davon ob es uns bewußt ist oder nicht,
5. Kapitel – ALLGEMEINE SYSTEMTHEORIE UND KYBERNETIK: thematisiert die Grundbegriffe des Bertalanffy-Programms, Evolution, Transformation, Selbstorganisation, Emergenz, Beer's Viable System Model für System-Diagnose, -Regulierung, -Design und liefert die methodischen Grundlagen für den Umgang mit komplexer Systeme
6. Kapitel – ONTOGENETISCHE ASPEKTE: thematisiert die kognitiv-psychologischen und gruppen-dynamischen Grundlagen für das collaborative Problemlösen in komplex-strukturierte DomainOntologien auf der Basis vom adaptiven Experiential Learning Model und pragmatischen Führungstheorien.
7. Kapitel – EMPIRISCHE THEORIEMODELLE UND SYSTEME – thematisiert die z.Zt. einzige erprobte Methode der Rekonstruktion erfahrungswissenschaftlicher Theorien, die als Grundlage für computerbasierte DomainOntologien dient.
8. Kapitel – SEMANTISCHE KONZEPTION VON THEORIEMODELLEN – thematisiert die wissenschaftstheoretischen Grundlagen für die problemorientierten Metamodellierung, d.h, Strukturen, Modelle, Potentielle Modelle, Empirische Systematisierung, Theoretizität, Partielle potentielle Modelle, Intendierte Systeme, Messung, Empirische Behauptung, Theorienetze, Spezialisierung, Theoretisierung, Reduktion, Querverbindungen, Theorie-Holone etc.
9. Kapitel – REPRÄSENTATION UND HANDLUNGSTHEORIE - thematisiert die handlungstheoretischen Grundlagen der Intaraktion in komplexen Systemen: Reflexion und Intentionalität, Handeln und Kommunikation, Begriffsnetze und Propositionen
10. Kapitel – PROBLEMLÖSEN MIT THEORIENETZEN - thematisiert das erfahrungswissenschaftliche Problemlösen als Planen mit Konflikte in Theorienetzen und liefert Algorithmen für die Implementierung. Anders als bisher bekannten Planungsansätzen kann man mit Theorienetzen keine abstrakte, sondern problembezogenen Plänen generieren.

11. Kapitel – ONTOGENETISCHE ANSÄTZE - thematisiert "Draw a distinction" als die elementare erkenntnistheoretischen Operation, worauf man Erfahrung und Erkenntnis, Situationen und Handlungen, Ontologien und charakteristische Verhaltensweisen gebunden sind. Die methodologische Systematisierung benutzt formale Begriffsanalyse, begriffliche Graphen, Persönliche Konstrukte, und Analogie-Modelle, um Ontologien, Metamodelle und Referenzmodelle, Laws of Form, Konzeptuelle Modelle, und Modelltransformationen.
12. Kapitel – ACCORD METHODOLOGIE - thematisiert die ACCORD-Methodologie, als Syntese all dieser Ansätze, d.h. ACCORD als Selbstorganisationsstruktur und generischer Rahmen für die Lösung komplexer Probleme.
13. Kapitel – ONTOENGINEERING ANSÄTZE - thematisiert die historische Entwicklung der Sprachen und Generationen: Niedere Programmiersprachen, Höhere Programmiersprachen, Datenbank- und Anwendungssprachen, Objektorientierte Programmiersprachen
14. Kapitel – OO-PARADIGMA MERKMALE - thematisiert Klassen und Objekte, Abstraktion, Datenkapselung, Vererbung, Wiederverwendbarkeit, Polymorphie, Vorgehensmodelle bei der OO – Softwareentwicklung, Anwendungsfälle und der BASE-Ansatz.
15. Kapitel – VON MODELLEN ZUR ANWENDUNG - thematisiert die bisher begründete Vorgehensweise: Strukturalistische Konzeptualisierung, Problemlösen mit AgentenNetzen, Verteiltes Problemlösen, Die Multi-Agenten Perspektive, Planen in Multiagenten-Systemen.
16. Kapitel – INTERNET UND CLIENT/SERVER - thematisiert Plattformen für verteiltes Problemlösen, Transaktionen, Server-Aspekte, Middleware usw.
17. Kapitel – COMPONENTEN PARADIGMA – thematisiert die komponentenbasierte Softwareentwicklung: komponentenbasierte Anwendungssysteme, legosteine-Metapher, flexible Verteilung und Anwendungsnahe, Schnittstellenspezifikation und Implementierung, Programmierung von Applikationen mit BC4J, Migration zu komponentenbasierten Softwaresystemen.
18. Kapitel – OBJEKTRELATIONALER DATENBANKDESIGN - thematisiert die Bedeutung von Datenbankmodell und –design, Objektmodell und Datenbank-Design, Regeln für die Abbildung Objektmodells auf relationale Implementierung, Die Architektur von Java Database Connectivity (JDBC), Typen von JDBC-Treibern und die JDBC-Treiber von Oracle.
19. Kapitel – DIE BC4J-FUNKTIONALITÄT - thematisiert die Business Components 4 Java: Konzeptuelle Modelle und Modelltransformationen, BC4J - alternative Einsatzbereiche usw.
20. Kapitel – VERTEILTEN ANWENDUNGEN - thematisiert J2EE vs. .NET - Unterschiede zwischen J2EE und .NET, Visual Studio .NET, Web Services, Web Portale, das ACCORD-Portal, Enterprise Application Integration (EAI), House of Business Engineering.
21. Kapitel – BUSINESS CAPABILITY MATURITY MODEL - thematisiert die Bewertungskriterien für die computerbasierte Unterstützung von Geschäftsprozesse: von Level 0 – zufällig, bis Level 4 – gemeistert
22. Kapitel – ACCORD UND GENERALPLAN VON UNTERNEHMEN - thematisiert die Modellierung von Organisationen und Softwaresystemen, Generalplan-Industrieunternehmen, Orientierung und Identifizierung der Funktionen, ACCORD und GPS Software Atlas.

23. Kapitel – ORGANISATORISCHER WANDEL - thematisiert den technologie Wandel, durch das Internet, die Informationstechnologie als Garant für Unternehmenserfolg, Produktivitätsparadoxons der Informationstechnik und ihre Wirtschaftlichkeitspotentiale.
24. Kapitel – ARCHITEKTURRAHMEN FÜR GESCHÄFTSPROZESSOPTIMIERUNG - thematisiert den Zyklus zum organisatorischen Wandel, die problemorientierte Differenzierung des abduktiv-deduktiven Zyklus, die praktische Anwendung des ACCORD-Modells im Handel, die methodische Anwendung des ACCORD-Vorgehensmodells.
25. Kapitel – DIGITALE FABRIK - thematisiert der IT-Einsatz im Unternehmen, Entwicklungsarten, Komplexität = ProduktOntologie x FabrikOntologie, Kriteriensystem und Technologiebewertung, Ontologiebasiertes Konzeptdesign, Quality Function Deployment (QFD), Fehlermöglichkeits und EinflußAnalyse (FMEA) und Vorgehensmodelle für Continuous Engineering.
26. Kapitel – EPILOG: SEMANTIC WEB – thematisiert die Internet-Zukunft.
27. Kapitel - LITERATUR

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.